

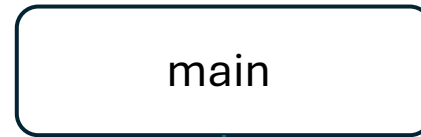
UAV Deconfliction

Design decision

Project Structure

- - config
 - config.py – Configuration parameters for UAVs and simulation.
 - src
 - main.py – Main entry point for the UAV deconfliction system.
 - collision_detection.py – Collision detection logic and utilities.
 - handle_input_data.py – Handles input data for primary UAV.
 - simulationUavDB.py – Simulation UAV data definitions.
 - waypoint_plot.py – 3D plotting of UAV paths.
 - uav_info_def
 - __init__.py – Package initializer for UAV info definitions.
 - uav_info.py – UAV data structures (position, time window, etc.).
 - test
 - test_main.py – Unit tests for main functionality.

Control flow



Pass primary UAV
data

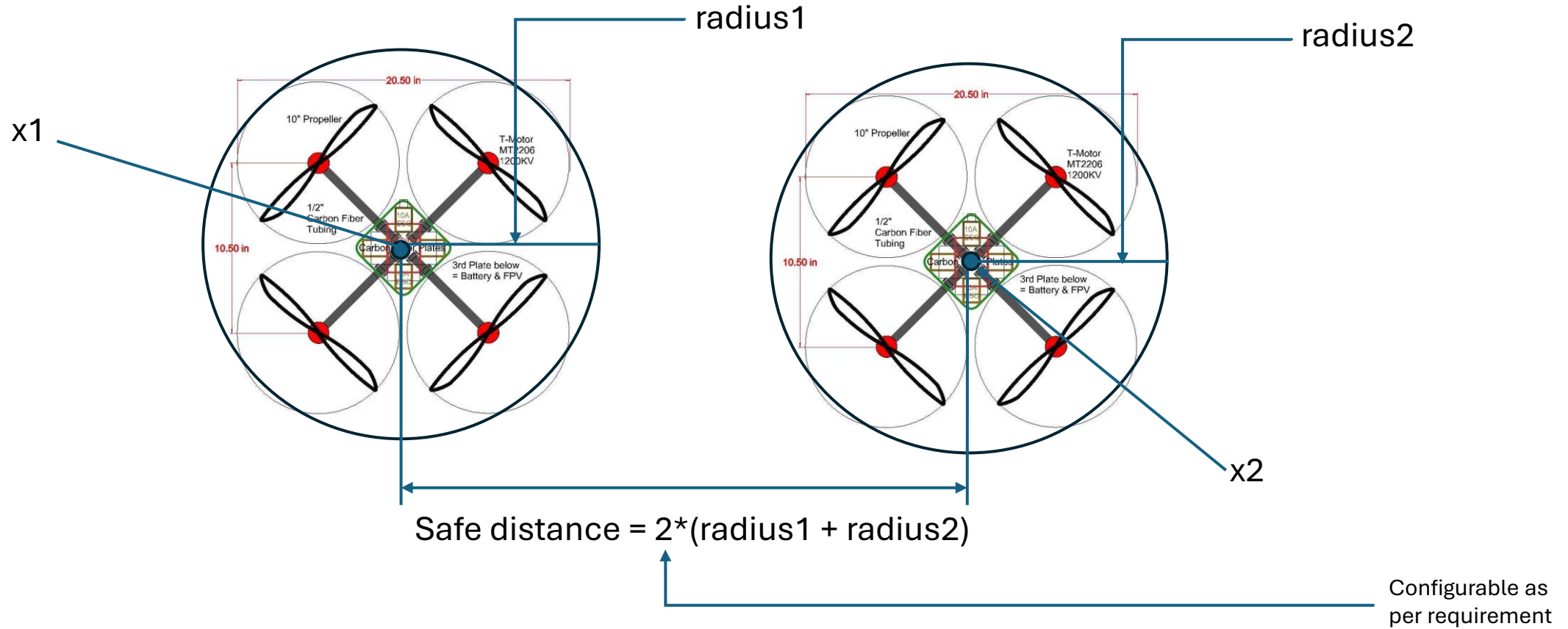
Handle_collision

- Detect_collision
 - Compare time window of primary UAV and Simulation UAV
 - If time window conflicts
 - Check if distance b/w primary and simulation UAV more than configured safeDistance
 - If distance b/w both UAVs less than safeDistance,
 - mark collision detection flag.
 - Print collision info
 - Print UAV info
- Plot all UAVs path

Handle_Inputdata

- Get primary UAV data

Calculating safe distance b/w two drones



$$(\text{abs}(x_1 - x_2) > \text{safe distance}) \text{ AND } (\text{abs}(y_1 - y_2) > \text{safe distance}) \text{ AND } (\text{abs}(z_1 - z_2) > \text{safe distance})$$